FUNCTIONAL SPECIFICATION

Multi-API Embedded Input Library

2017

By James Schmidt

Version History

Version 1.0 – Draft

FUNCTIONAL SPECIFICATION –
Multi-API Embedded Input Library

Table of Contents

1. INTRODUCTION, Overview, and Purpose

This library will allow programmers the ease of having a robust input handling system without having to deal with the difficulty of hooking up all the necessary functionality. The majority of API's only give a basic on or off (pressed or not pressed) event for programmers to use. This library will hook into those multiple API's and give programmers a higher level of input functionality. There will also be example code and demos for more advanced input features that programmers can utilize for their engines/games. The final project will be a library that can be added to any API project/game on any Windows 7 or higher PC.

1.1 Outcomes and Scope

The planned outcome of this project is a library that can be embedded for use with the following API's:

- SDL
- SFML
- GLFW
- Windows API with XInput

1.2 Project Requirements

- The project should be able to work with the API's listed in 1.1.
- The project should be detailed and easy for programmers to understand/use.
- The project should come with example code and demos for programmers to understand how the features are utilized.

1.3 Development Requirements

- The development environment for this project will be Visual Studio 2015.
- Perforce (P4V) will be used for source control.
- Cross-Platform integration will not be a focus of this library, even though some of the API's can handle multiple operating systems.

1.4 General Constraints

- The library must work on any Window 7 or higher PC.
- The library must compile properly and work with Visual Studio 2015.
- Every feature of the library must work with every API listed in 1.1.

2. FUNTIONAL DESCRIPTION

2.1 User Functionality

Programmers should be able to read the provided example code and demos to gain proper insight on how the library is used. Detailed function headers will grant programmers with more in-depth information.

3. SPECIFIC REQUIREMENTS

3.1 External Interface Requirements

The library must work with the following peripherals:

- Keyboard
- Mouse
- Joystick

3.2 Library Features

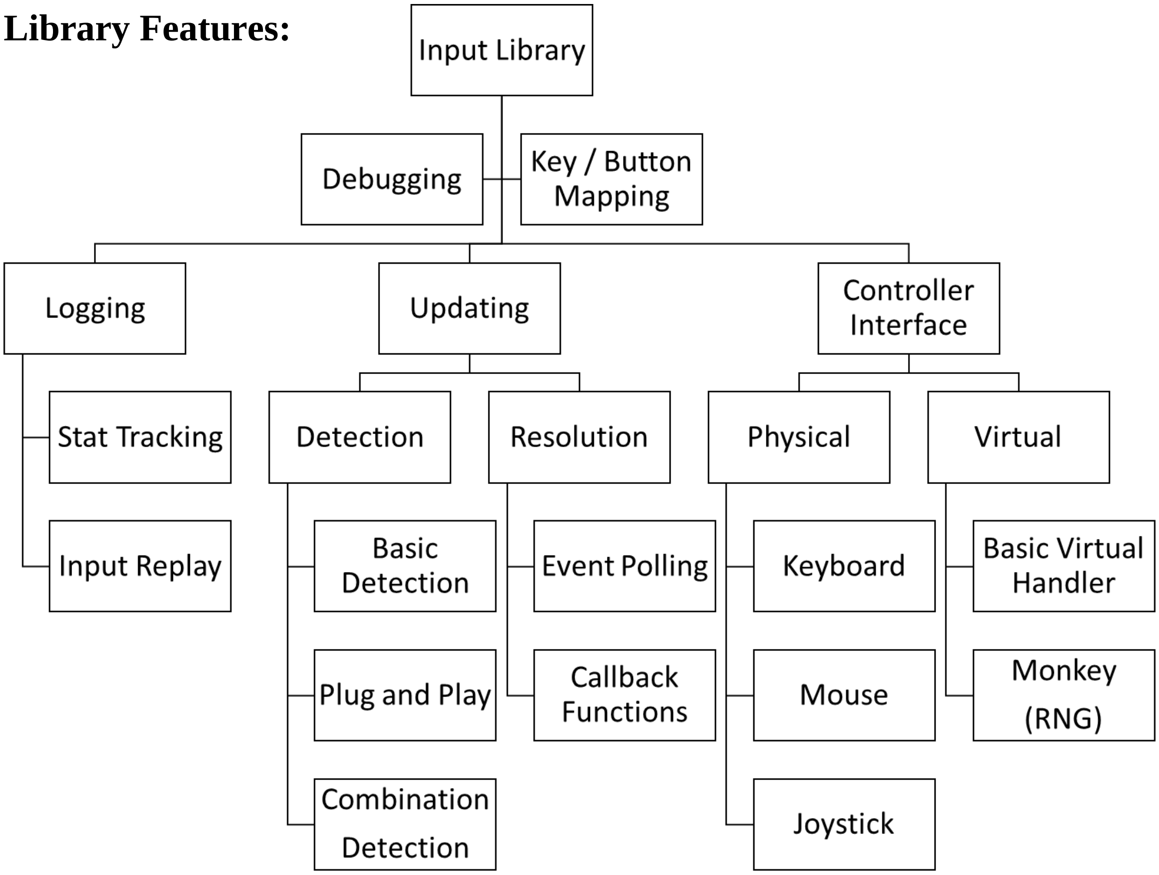
- Debugging
- Key/Button Mapping
- Stat Tracking
- Input Detection
- Plug and Play
- Combination Detection
- Virtual Controller
- Monkey

3.2 Advanced Functionality

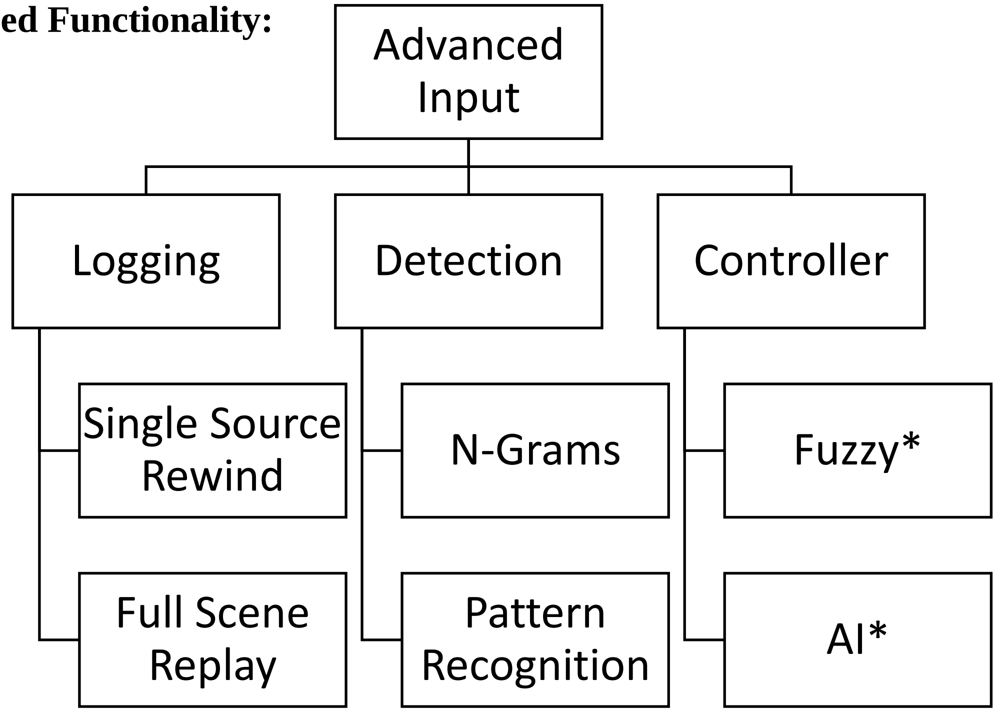
The advanced functionality list below will not be built directly into the library due to their implementations being dependent on other systems outside of input. However, there will be example code and demos for how they can be implemented using the functionality provided by the library.

- Replay
- Single Source Rewind
- N-Grams
- Pattern Recognition
- Fuzzy Controller*
- AI Controller*

Library Features:



Advanced Functionality:



* scope and time dependent